

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 27, 2011

B. Carpenter
Univ. of Auckland
S. Jiang
Huawei Technologies Co., Ltd
Z. Cao
ChinaMobile
February 23, 2011

Problem Statement for Referral
draft-carpen-ter-referral-ps-02

Abstract

The purpose of a referral is to enable a given entity in a multiparty Internet application to pass information to another party. It enables a communication initiator to be aware of relevant information of its destination entity before launching the communication. This memo discusses the problems involved in referral scenarios.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

Internet-Draft

Referral Problem

February 2011

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Goals of Referral	4
3.1.	Reachability	4
3.2.	Path Selection	4
3.2.1.	An Example: Triangle Path Optimization	4
3.3.	Interface Selection	5
4.	Problem Statement	6
4.1.	IP Addresses are not sufficient	6
4.2.	FQDNs are not sufficient	7
4.3.	Relevant Information is lacking	9
4.4.	Extra complexity from ID-Locator Split Mechanisms	9
5.	A Generic Referral Mechanism is needed	10
6.	Security Considerations	11
7.	IANA Considerations	12
8.	Acknowledgements	12
9.	Change log	12
10.	Informative References	12
	Authors' Addresses	14

Internet-Draft

Referral Problem

February 2011

1. Introduction

A frequently occurring situation is that one entity A connected to the Internet (or to some private network using the Internet protocol suite) needs to be aware of the information of another entity B in order to reach it. The information can be obtained from B itself or some third-party entity C. This is known as a referral.

Referral is the act whereby one entity informs another entity how to contact a specific entity. It enables a communication initiator to be aware of relevant information of its destination entity in order to launch a communication channel. This referral information can be obtained through an existing communication channel between these two entities or from third-party entities.

In the original design of the Internet, IP addresses were global, unique, and quasi-permanent. Also any differentiation beyond that provided by an IP address was done by protocol and port numbers. Referrals were therefore performed simply by passing an IP address and possibly protocol and port numbers. In fact simple referrals (the first case above, sometimes called first-party referrals) were never needed since A could simply use B's address. Third-party referrals were trivial: C would tell A about B's address. Thus, it became common practice to pass raw addresses between entities. A classical example is the FTP PORT command [[RFC0959](#)].

2. Terminology

This document makes use of the following terms:

- o "Entity": we use this rather than "application" to describe any software component embedded in an Internet host, not just a specific application, that sends, receives or makes use of referrals. Also, in case of dynamic load sharing or failover, an entity might even migrate between hosts.
- o "Referral": the act of one entity informing another entity how to

- contact a specific entity.
- o "Reference": the actual data (name, address, identifier, locator, pointer, etc.) that is the basis of a referral.
- o "Referring entity": the entity that sends a referral.
- o "Receiving entity": the entity that receives a referral.
- o "Referenced entity": the target entity of a reference.
- o "Scope": the region or regions of the Internet within which a given reference is applicable to reach the referenced entity.

[3.](#) Goals of Referral

The principal purpose of referral is to enable one entity in a multi-party application to pass information to another party involved in the same application. This document makes no assumptions about whether the entities are acting as clients, servers, peers, super-nodes, relays, proxies, etc., as far as the application is concerned. Neither does it take a position as to how the various entities become aware of the need to send a referral; this depends entirely on the structure of the application.

[3.1.](#) Reachability

The primary goals of referral is to enable a communication initiator to reach its destination entity. Referral is a best effort mechanism. It does not guarantee actual reachability, since the referring entity has no general way of knowing which paths exist between the receiving entity and the referenced entity. Even if a reference is theoretically in scope, and within its defined lifetime, it may have become unreachable since it was sent. A receiving entity should always be prepared for reachability failures and associated retry and failover mechanisms, which are out of scope for the referral mechanism itself.

[3.2.](#) Path Selection

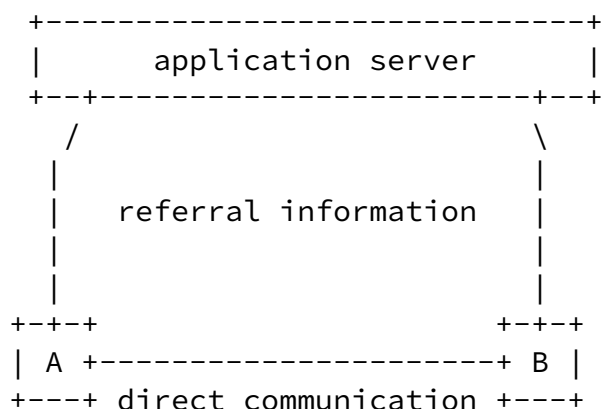
A reference might carry multiple references for the same target. These may lead to multiple possible paths from the receiving entity to the referenced entity. This scenario is particularly generic when

the destination or/and source entity has multiple interfaces or is multi-homed.

The referring entity is not likely to know which path is best. The receiving entity will need to make a choice, possibly by local policy (e.g. [RFC3484]) or possibly by trial and error (e.g. [RFC4038], [RFC5534]). This choice is also out of scope for the referral mechanism itself.

3.2.1. An Example: Triangle Path Optimization

In application scenarios, the triangular path shown below is common. Both Host A and Host B connect to an application server and the application server forwards traffic as a relay agent. A slightly more complicated scenario is when the two hosts connect to different application servers individually and application servers talk to each other's relay agents. In SIP, this is often called the "SIP trapezoid".

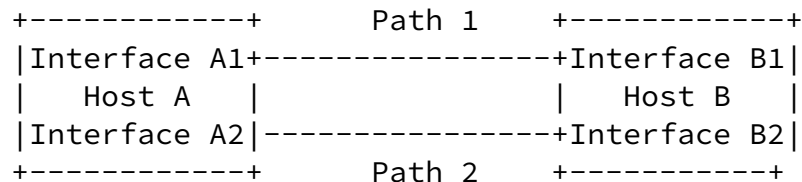


By passing A's reference to B, B can try to communicate directly with A, using the communication line at the bottom. If the direct communication is established successfully, the triangle path gets optimized. Both the application server and network bandwidth can be benefit from this operation.

3.3. Interface Selection

We also encounter multi-interfaced hosts whose reachability is bound to a particular (logical/physical) interface. More information is

required to indicate which interface may be used under different circumstances. The multi-interface problem is defined and studied by the IETF MIF WG. Here referral can provide host A's multi-interface information to host B; accordingly, host B can select one of the interfaces to establish a connection.



For example, as shown in the above figure, Host A has connected to Host B through Path 1. They can exchange references through Path 1. They may discover that Path 2, using different interfaces, is better than Path 1, maybe cheaper, faster or more stable. Then, they can switch to Path 2. For example, Host A has interface A1 as broadband access, almost free; and interface A2 is 3G access, which costs 0.1 \$ per MB. Both of them are available for incoming connections. If this information is passed to host B, through referral, then host B should choose the A1 interface to reach host A. Such information is useful to express a host's status or preference.

In order to choose between different interfaces, not only the connectivity information of these interfaces, but also some additional

information may be helpful, such as bandwidth, financial cost, latency, etc. This additional information may also be provided through referral. However, this additional information, even if accurate when sent by the referring entity, may nevertheless be invalid at the location of the receiving entity.

4. Problem Statement

Unfortunately, the simple approach to referrals, passing an IP address, often fails in today's Internet. As has been known for some time [[RFC2101](#)], hosts' IP addresses no longer all have global scope. They often have limited reachability, and may have limited lifetime. They are not sufficient to establish communication in many cases of dynamic referrals, for a variety of reasons. FQDNs may be used

instead in some scenarios. However, FQDNs also have their own limitations and may fail in some scenarios.

4.1. IP Addresses are not sufficient

It is no longer reasonable to assume that a host with a fixed location has a fixed IP address, or even a stable IP address.

Furthermore, in the context of IPv4 address exhaustion, several solutions have emerged to share a single public IPv4 address between several customers simultaneously. Consequently, a public IPv4 address often no longer identifies a single customer/user/host, while a private IPv4 address is meaningless out of the private network scope. Other information (e.g., port range) is required to identify unambiguously a given customer/user/host. Both IP addresses and port numbers may be different on either side of a NAT or some other middlebox [[RFC3234](#)], and firewalls may block them. It is no longer reasonable to assume that an IP address for a host, which allows a given peer to reach that host in one location, also works from a different location - even if that host is reachable from the second location.

Also, the Internet now has two co-existing address formats for IPv4 and IPv6. Direct communication can only be established when both peers use the same IP version. Having the address of the source and destination in the same IP version does not necessarily mean that the path will be using that IP version. Simple approaches may cause unnecessary double translation [[I-D.boucadair-softwire-cgn-bypass](#)]. Some addresses may even be the result of translation between IPv4 and IPv6, with severe limitations on their scope and lifetime. Sending an out-of-scope or expired address, or one of the wrong format, as a referral will fail.

A specific problem of this kind may be caused by NAT64 [[I-D.ietf-behave-v6v4-xlate-stateful](#)]. If an IPv6-only host behind a NAT64 obtains a synthetic IPv6 address for an IPv4-only host, it can communicate successfully via the NAT64. However, if the synthetic address is referred to another IPv6 host, it may or may not work correctly. We can consider four cases:

1. If the receiving entity is behind the same NAT64 as the referring

- entity, all should be well.
2. If the referring entity and the receiving entity are behind different NAT64 devices, both using the defined Well Known Prefix [[RFC6052](#)], all should be well, because the same synthetic address will work in both cases.
 3. If the receiving entity is behind a different NAT64 that uses a Network Specific Prefix [[RFC6052](#)], the synthetic address will be meaningless and communication will fail. The only way to avoid this failure is for the original NAT64's Network Specific Prefix to be globally reachable, which seems highly unlikely for operational and security reasons.
 4. If the receiving entity is a dual stack node that is not behind a NAT64, the synthetic address will be meaningless. Although there is an IPv4 path to the target host, the receiving entity will not know how to find it. Again, the only way to avoid this failure is for the original NAT64's prefix to be globally reachable.

In the last two cases, even if connectivity failure is avoided, the path taken by the packets will be far from optimal, traversing the original NAT64.

IP addresses today may have an implied "context" (VPN, VoIP VC, IP TV, etc.): the reachability of such an address depends on that context.

An implication of these issues is that there is no clean definition of the scope of an address (especially an IPv4 address, due to the prevalence of NAT). It is impossible to determine algorithmically, by inspecting the bits of an address, what its scope of reachability is. Resolving this problem would greatly clarify the general problem of referrals.

[4.2.](#) FQDNs are not sufficient

In some cases, this problem may be readily solved by passing a Fully Qualified Domain Name (FQDN) instead of an IP address. Indeed, that is an architecturally preferred solution [[RFC1958](#)]. However, it is not sufficient in many cases of dynamic referrals. Experience shows that an application cannot use a domain name in order to reliably find usable address(es) of an arbitrary peer. Domain names work

fairly well to find the addresses of public servers, as in web

servers or SMTP servers, because operators of such servers take pains to make sure that their domain names work. But DNS records are not as reliably maintained for arbitrary hosts such as might need to be contacted in peer-to-peer applications, or for servers within corporate networks. Many small networks do not even maintain DNS entries for their hosts, and for some networks that do list local hosts in DNS, the listings may well be unusable from a remote location, because of two-faced DNS, or because the A record contains a private address. These cases may even be intentional as part of a security ring-fence, where w3.example.com only resolves within the corporate boundary, and/or resolves to IP addresses which are only reachable within the corporate administrative boundaries. In such contexts, incoming connections are usually filtered by the corporate firewall.

An additional issue with FQDNs is the very common situation where multiple hosts are hidden behind a NAT, but they share one FQDN which is in fact a dummy name, created automatically by the ISP so that reverse DNS lookup will succeed for the NAT's public IPv4 address. Such FQDNs are useless for identifying specific hosts.

Furthermore, an FQDN may not be sufficient to establish successful communications involving heterogeneous peers (i.e., IPv4 and IPv6) since A and AAAA records may not be consistently provisioned. There are known cases where a server has one name that produces an A record (e.g., www.example.com) and another name that produces an AAAA record (e.g., ipv6.example.com). An additional complication is that some answers from DNS may be synthetic IP addresses, e.g., AAAA records sent by DNS64. The host may have no means to detect that such an address represents an IPv4 host. These addresses should not be interpreted as native IPv6 address.

In such cases, an IP address either cannot be derived from an FQDN, or if so derived, cannot be accessed from an arbitrary location in the Internet.

A related problem is that an application does not have a reliable way of knowing its own domain name - or to be more precise, a way of knowing a domain name that will allow the application to be reached from another location.

There are wider systemic problems with the DNS as a reliable way to find a usable address, which are somewhat out of scope here, but can be summarised:

- o In large networks, it is now quite common that the DNS administrator is out of touch with the applications user or administrator, and as a result, that the DNS is out of sync with

- reality.
- o DNS was never designed to accommodate mobile or roaming hosts, whose locator may change rapidly.
 - o DNS has never been satisfactorily adapted to isolated, transiently-connected, or ad hoc networks.
 - o It is no longer reasonable to assume that all addresses associated with a DNS name are bound to a single host. One result is that the DNS name might suffice for an initial connection, but a specific address is needed to rebind to the same peer, say, to recover from a broken connection.
 - o It is no longer reasonable to assume that a DNS query will return all usable addresses for a host.
 - o Hosts may be identified by a different URI per service: no unique URI scheme, meaning no single FQDN, will apply.

For all the above reasons, the problem of address referrals cannot be solved simply by recommending the use of FQDNs instead. The guideline in [[RFC1958](#)] is in fact too simple for today's network. Something more elaborate than an IP address or an FQDN appears to be needed in the general case of application referrals.

[4.3.](#) Relevant Information is lacking

Neither an IP address nor an FQDN gives complete information about the referenced entity. For example, IP addresses normally have associated lifetimes (derived from DHCP, SLAAC or the relevant DNS TTL), so they should be treated as invalid after their lifetimes expire. A referral that does not convey the lifetime associated with an address is problematic. As mentioned above, the scope of a reference also affects its usefulness. These are examples of additional information that is necessary to correctly interpret a referral; therefore part of the problem is conveying such information along with the reference.

[4.4.](#) Extra complexity from ID-Locator Split Mechanisms

Additional complexity for referrals would come from the deployment of any technology that separates locators from identifiers, rather than combining the two as an IP address. Since a very wide range of such solutions have been proposed (e.g. HIP, LISP, ILNP and Name-based Sockets) [[I-D.ubillos-name-based-sockets](#)], it is difficult to define the resulting problems precisely.

However, to consider the example of Name-based Sockets, if a referral was made based on the IP address being used at a given instant for a Name-based Socket, that address might be useless by the time the

referral was completed, because the socket suddenly migrated to a different IP address.

The SHIM6 protocol [[RFC5533](#)] and the Multiple Interfaces (MIF) Working Group may produce similar difficulties, since they also consider scenarios where the IP address in use for some purpose may change unexpectedly.

Any referral mechanism must be able to deal with situations where the locator corresponding to a given identifier is subject to change.

[5.](#) A Generic Referral Mechanism is needed

The first motivation is the observation that unless the parties involved have reached an understanding about the scope, lifetime, and format of the elements in a referral through some other means, that information must be passed with the referral. This is required so that the receiving entity can determine whether or not the referral is useful. The referral therefore needs to consist of a fully-fledged data structure, or to be made using a mutually agreed referral protocol.

When an attempt to establish a communication channel based on certain referral information fails, good design suggests that the receiving entity should attempt to correct the situation. For example, if communication fails to be established using an IP address, it would often be appropriate to attempt a DNS lookup, despite the difficulties mentioned above. The second motivating problem is that it may be helpful to the entity receiving a reference to also receive information about the source of the reference, such as an FQDN, if that is known to the sender of the reference. The receiving entity can then attempt to recover a valid address (and possibly port number) for the referred entity.

The third motivating problem is to allow a reference to contain alternatives to an IP address or an FQDN, when any such alternatives exist.

Additional arguments for a generic referral mechanism include:

1. Allow for general mechanisms that can be used by any application to handle references and understand the meaning of referral

information, such as IP address, possibly protocol and port numbers. However, there is an open question whether this standard referral design should be used for new applications only, or extended to existing applications.

2. Simplify ALG design during middlebox traversal. There are middleboxes, like firewalls and translators, especially in the mobile network, which require application layer gateways ALG. The cost of ALG functions is huge for the mobile operator in terms of implementation, performance. Standard references could

simplify ALG implementation during middlebox traversal in the mobile network.

3. Simplify packet inspection. Operators sometimes need to inspect information or details during communication for administration reasons. If referral mechanism is standardized, it is easier for an operator to capture and investigate the required information.

We observe that we have identified two general requirements: the need to define address scope more precisely, and the need to communicate references in a generic way.

It should be noted that partial or application-specific solutions to these problems abound, because any multi-party distributed application must solve them. The best documented example is ICE [[RFC5245](#)], which is an active protocol specific to applications mediated by SDP [[RFC4566](#)]. ICE "works by including a multiplicity of IP addresses and ports in SDP offers and answers, which are then tested for connectivity by peer-to-peer connectivity checks." The question raised here is whether we can define requirements for a generic solution that can be used by future applications, and possibly be retro-fitted to existing applications.

One approach could be a "SuperICE" designed to be completely general and not tied to the SDP model. Another approach is the idea of a generic referral object. Such an object could be passed between the entities of a multi-party application, but without defining a specific protocol for that purpose. Some applications might choose to send it in-band as a raw binary object, others might use a simple ASCII encoding, and still others might prefer to encode it in XML, for example. Finally, it might also be used as part of SuperICE.

[6.](#) Security Considerations

It should be noted that referral should not function as a way to nullify the effect of a firewall or any other security mechanism. If the receiving entity chooses a particular reference and attempts to send packets to the corresponding IP address, whether they are delivered or not will depend on the existing security mechanisms, whatever they may be.

Nevertheless, if a site security policy requires it, certain references may be excluded from referral information sent to certain destinations. This would require a security policy mechanism to be added to the process of generating referral information.

Forged or intercepted referral information would enable a wide variety of attacks. Although not fundamentally different from

attacks based on forged or observed IP addresses or FQDNs, no doubt referral would allow such attacks to be more ingenious, simply because they provide more information than an address or FQDN alone. Referral information should be transmitted through authenticated and encrypted channels. It is not further elaborated here.

Referral may raise potential privacy issues, which are not explored in this document. For example, in the SIP context, mechanisms such as [\[RFC3323\]](#) and [\[RFC5767\]](#) are available to hide information that might identify end-points. Referral usage scenarios must ensure that they do not unintentionally defeat privacy solutions.

[7.](#) IANA Considerations

This document requests no action by IANA.

[8.](#) Acknowledgements

Bo Zhou, formerly with China Mobile, was an original author of this document. His contributions are gratefully acknowledged.

Valuable comments and contributions were made by Mohamed Boucadair, Dan Wing, Keith Moore and others.

This document was produced using the xml2rfc tool [[RFC2629](#)].

9. Change log

[draft-carpenter-referral-ps-00](#): original version, 2010-06-21.

[draft-carpenter-referral-ps-01](#): add content regarding to ID-Locator Split Mechanisms, 2010-08-30.

[draft-carpenter-referral-ps-02](#): add content regarding NAT64, 2011-02-23.

10. Informative References

[I-D.boucadair-softwire-cgn-bypass]
Boucadair, M., Jacquenet, C., Song, J., and Q. Niu,
"Procedure to bypass DS-Lite AFTR",
[draft-boucadair-softwire-cgn-bypass-03](#) (work in progress),
October 2010.

Carpenter, et al. Expires August 27, 2011 [Page 12]

Internet-Draft Referral Problem February 2011

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful
NAT64: Network Address and Protocol Translation from IPv6
Clients to IPv4 Servers",
[draft-ietf-behave-v6v4-xlate-stateful-12](#) (work in
progress), July 2010.

[I-D.ubillos-name-based-sockets]
Ubillos, J., Xu, M., Ming, Z., and C. Vogt, "Name-Based
Sockets Architecture", [draft-ubillos-name-based-sockets-03](#)
(work in progress), September 2010.

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol",
STD 9, [RFC 959](#), October 1985.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet",
[RFC 1958](#), June 1996.

- [RFC2101] Carpenter, B., Crowcroft, J., and Y. Rekhter, "IPv4 Address Behaviour Today", [RFC 2101](#), February 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", [RFC 2629](#), June 1999.
- [RFC2775] Carpenter, B., "Internet Transparency", [RFC 2775](#), February 2000.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", [RFC 3234](#), February 2002.
- [RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", [RFC 3323](#), November 2002.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", [RFC 3484](#), February 2003.
- [RFC4038] Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E. Castro, "Application Aspects of IPv6 Transition", [RFC 4038](#), March 2005.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.

Carpenter, et al.

Expires August 27, 2011

[Page 13]

Internet-Draft

Referral Problem

February 2011

- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", [RFC 5533](#), June 2009.
- [RFC5534] Arkko, J. and I. van Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", [RFC 5534](#), June 2009.
- [RFC5767] Munakata, M., Schubert, S., and T. Ohba, "User-Agent-Driven Privacy Mechanism for SIP", [RFC 5767](#), April 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X.

Li, "IPv6 Addressing of IPv4/IPv6 Translators", [RFC 6052](#),
October 2010.

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing
P.R. China

Email: jiangsheng@huawei.com

Zhen Cao
China Mobile
Unit2, 28 Xuanwumenxi Ave, Xuanwu District
Beijing, 100053
P.R. China

Email: caozhenpku@gmail.com